ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER Applied Mathematics and Computation xxx (2005) xxx–xxx

APPLIED MATHEMATICS AND COMPUTATION

www.elsevier.com/locate/amc

An efficient convergent lattice algorithm for European Asian options

Tian-Shyr Dai [a,*], Guan-Shieng Huang [b], Yuh-Dauh Lyuu [c,d,1]

[a] *Department of Applied Mathematics, Chuan-Yuan Christian University, 22 Pu-Jen, Pu-chung Li, Chung-Li, Tao-Yuan county 32023, Taiwan*

[b] *Department of Computer Science and Information Engineering, National Chi Nan University, Taipei 545, Taiwan*

[c] *Department of Finance, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei 106, Taiwan*

[d] *Department of Computer Science and Information Engineering, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei 106, Taiwan*

Abstract

Financial options whose payoff depends critically on historical prices are called path-dependent options. Their prices are usually harder to calculate than options whose prices do not depend on past histories. Asian options are popular path-dependent derivatives, and it has been a long-standing problem to price them efficiently and accurately. No known exact pricing formulas are available to price them under the continuous-time Black–Scholes model. Although approximate pricing formulas exist, they lack accuracy guarantees. Asian options can be priced numerically on the lattice. A lattice divides the time to maturity into n equal-length time steps. The option price computed by the lattice converges to the option value under the Black–Scholes model as $n \to \infty$. Unfortunately, only subexponential-time algorithms are available if Asian options are to be priced on the

* Corresponding author.

E-mail address: cameldai@cycu.edu.tw (T.-S. Dai).

[1] The author was supported in part by NSC grant 92-2213-E-002-016.

0096-3003/$ - see front matter
doi:10.1016/j.amc.2004.10.085

lattice without approximations. Efficient approximation algorithms are available for the lattice. The fastest lattice algorithm published in the literature runs in $O(n^{3.5})$-time, whereas for the related PDE method, the fastest one runs in $O(n^3)$ time. This paper presents a new lattice algorithm that runs in $O(n^{2.5})$ time, the best in the literature for such methods. Our algorithm exploits the method of Lagrange multipliers to minimize the approximation error. Numerical results verify its accuracy and the excellent performance.

Keywords: Option pricing; Lattice; Path-dependent derivative; Asian option; Approximation algorithm; Lagrange multiplier

1. Introduction

Derivative securities are financial instruments whose values depend on some underlying assets. Such securities are essential to speculation and the management of financial risk. Options are financial derivatives that give their buyers the right but not the obligation to buy or sell the underlying assets for a contractual price called the exercise price. Take the typical stock option for example. Assume that an investor purchases a call option, which gives him the right to buy 100 shares of XYZ stock at $10 per share 60 days from now. If the stock price ends above $10 then, say $25, then the buyer can realize a profit of $100 \times (25 - 10) = 1500$ dollars by exercising the option. If the stock price ends below $10, the buyer simply gives up the option. The payoff of this call option is therefore $100 \times \max(S - 10, 0)$, where S is the stock price 60 days from now. Note that S is a random variable. This option is commonly called a vanilla option for its simplicity.

In practice, many varieties of complex options have been structured to meet specific financial goals. Take path-dependent options as an example. A path-dependent option is an option whose payoff depends nontrivially on the price history of the underlying asset, which we will assume to be stock for convenience. The payoff function may depend on the maximum stock price, the minimum stock price, or the average stock price, to mention just a few possibilities. It may also depend on whether the stock price ever hits a given target price, whether the stock price ever stays within two given target prices for a given length of time, and so on. The possibilities are clearly without limits.

How to assign a fair price to an option given a continuous-time stochastic process for the stock price has been investigated since as early as 1900 [1]. In 1973, Black and Scholes [2] settle the question for vanilla option pricing in a way that is considered intellectually satisfactory. Although an option must have a unique theoretical price, calculating that price may be computationally difficult if the payoff is complicated. For example, Chalasani et al. show that the general path-dependent option-pricing problem is #P-hard [3].

This paper focuses on a particular type of path-dependent option, the Asian option, that is known to be difficult to price. Asian options seem to be suggested first by Ingersoll [4]. They were originally traded on Asian markets, particularly in Tokyo [5]. The payoff of an Asian option depends on the average price of the underlying asset. It is useful for hedging transactions whose cost is related to the average price of the underlying asset (such as crude oil). Its price is furthermore less subject to price manipulation. Hence the averaging feature is popular in many thinly-traded markets and embedded in other derivatives like convertible bonds.

There are no simple exact closed-form formulas for the price of Asian option under the standard continuous-time Black–Scholes model. Call this price the true option value for simplicity. Many approximate closed-form solutions have been proposed under various assumptions [6–8]. Geman and Yor derive an analytical expression for the Laplace transform of the Asian call option [9]. Numerical inversion of this transform is also considered in [10,11]. Some inversion algorithms based on the Euler and Post-Widder methods can be found in [12]. Rogers and Shi provide lower and upper bounds [13]. These formulas are surveyed in [14–16,5]. They show rather conclusively that most approximate closed-form formulas lack the accuracy guarantees and some even produce large pricing errors under certain circumstances.

Since no exact closed-form formulas exist for the Asian option, the development of efficient numerical algorithms becomes critical. To begin with, there are the popular Monte Carlo and quasi-Monte Carlo methods [17–21]. The main problem is their relative inefficiency.

The option value can be approximated by numerical methods such as the lattice and the related discretized PDE methods. These methods divide the time horizon of the option into n discrete time steps and discretize the stock prices at each time step. Take a 2-time-step CRR lattice model in Fig. 1 as an example. (The CRR lattice will be described in more detail later.) The time interval is evenly divided into 2 time steps. The stock price at time step 0 is S_0 (at node $N(0,0)$). The stock price can either move up to S_0u (at node $N(1,0)$) or down to S_0d (at node $N(1,1)$) at the first time step. Similarly, each stock price can either move up or move down in subsequent time steps. Discretization error is introduced by the CRR lattice model because both the time and the possible stock prices are discretized. Since the discretization error goes to zero at rate $O(n^{-1})$ [22], the option values computed by the CRR model converge to the true option value. The remaining key issue is whether such convergence can be achieved efficiently.

To see intuitively why pricing the Asian option on the lattice can be so time-consuming, assume that the random walk for the stock price is binomial as in Fig. 1. After n time steps, the history contains 2^n possible price paths, each with its own average stock price. As the payoff of the Asian option depends on the average stock price, there are 2^n possible payoffs at time step n. To price an

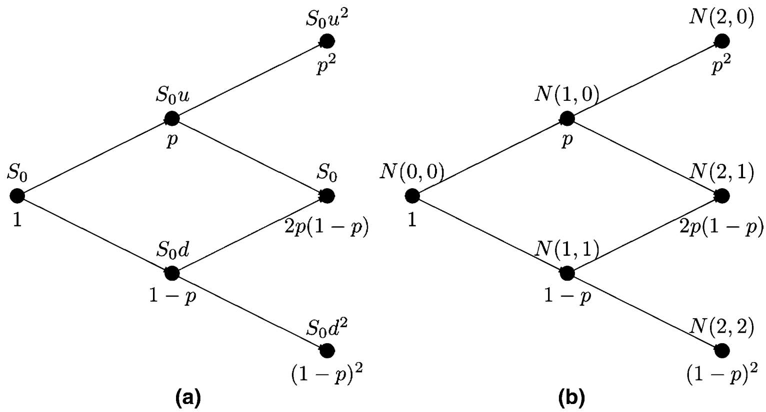

Fig. 1. A 2-time-step CRR lattice. (a) The stock price is placed above each node. (b) The node name is above each node. The probability of reaching each node from the root is labelled under the nodes in both plots.

Asian option on a lattice without introducing errors other than the discretization error, the most efficient algorithm currently known is by Dai and Lyuu [23,24]. But their algorithm runs in time subexponential in n.

In pricing the Asian option, each node on the lattice should keep a state for each possible average stock price. Then the option value corresponding to each state is calculated. The trouble with this straightforward approach is the exponential nature of the number of states. To strike a better balance between efficiency and accuracy, approximation algorithms usually allow errors besides the discretization error. The approximation algorithm proposed by Hull and White [25] employs much fewer states for each node (called the allocated states). Only the option value corresponding to an allocated state is evaluated. The option value corresponding to a missing state, in contrast, is interpolated from those of the two nearest allocated states. Interpolation errors are thus introduced. This influential paradigm has been followed by most approximation lattice algorithms [26–28].

The major problem with the Hull–White paradigm is convergence: Forsyth et al. show that the calculated option values may not converge to the true option value if the lattice algorithms are improperly implemented [22]. Efficient and convergent approximation algorithms on the lattice are available. For example, Aingworth et al. produce an $O(n^4)$-time algorithm with convergence rate $O(n^{-1})$ [29]; Dai et al. improve their running time from $O(n^4)$ to $O(n^{3.5})$ [30]; Forsyth et al. present an $O(n^{3.5})$-time approximation lattice algorithm and an $O(n^3)$-time discretized PDE method that both converge at rate $O(n^{-1})$ [22].

The major contribution of our paper is a new approximation lattice algorithm with a running time of $O(n^{2.5})$, the best in the literature. The convergence rate is $O(n^{-1})$. The true option value can be tightly estimated by extrapolation. Two key ideas are exploited in the algorithm. First, the option values of many states can be evaluated exactly by a simple formula without resorting to interpolation [29]. This dramatically reduces the interpolation errors accumulated during backward induction. The algorithm therefore focuses the computational efforts on the states that cannot be evaluated directly. The second idea is to allocate the number of states in such a way that the interpolation error can be minimized. This idea is pioneered by Dai et al. [30]. Intuitively, the states should be distributed based on the importance of each node. Technically, the distribution of states is calculated by applying the method of Lagrange multipliers to minimize the interpolation error. The application of Lagrange multipliers in option pricing is novel and makes the analysis rigorous.

The paper is organized as follows. The stock price dynamics is described in Section 2. How to price Asian options on the lattice and the efficiency problems are also dealt with in the same section. Section 3 presents our efficient approximation algorithm and proves the performance and convergence rate claims. Numerical results are given in Section 4 to support these claims. Section 5 concludes this paper.

2. Model, lattice, and pricing

Assume the Asian option initiates at 0 (in year) and matures at T (in years). Define $S(t)$ as the stock price at year t. $S(t)$ follows the log-normal diffusion process:

$$S(t+\mathrm{d}t) = S(t)\exp[(r-0.5\sigma^2)\,\mathrm{d}t + \sigma\mathrm{d}W_t], \tag{1}$$

where W_t is the standard Wiener process, r is the risk-free interest rate per annum, and σ denotes the volatility of the stock price.

The payoff of an Asian option depends on the average stock price at maturity defined as $A_T \equiv \frac{\int_0^T S(\tau)\mathrm{d}\tau}{T}$. Let X be the exercise price. The payoff of an Asian call option at maturity date is $\max(A_T - X, 0)$. The value of an Asian call option is therefore $\mathrm{e}^{-rT}E[\max(A_T - X, 0)]$ under the so-called risk-neutral probability measure. This paper focuses on Asian call options; the extension to Asian put options is straightforward.

Define the value of an Asian call option at year t as $V(S, A, t)$, where S and A denote the stock price at year t and the average stock price from year 0 to year t, respectively. $V(S, A, t)$ satisfies the following partial differential equation [26]:

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 V}{\partial S^2} + \frac{S-A}{t}\frac{\partial V}{\partial A} - rV = 0. \tag{2}$$

The above equation can be numerically solved by either the finite difference method or the lattice method. Both are discrete-time algorithms, which partition the time between year 0 and year T into n equal-length time steps. The length of a time step Δt is therefore T/n. Let S_i denote the stock price at time step i, which corresponds to $S(i\Delta t)$ in the continuous-time model. The average stock price from time step 0 to time step i in a discrete-time model is defined as $A(i) \equiv \frac{S_0+S_1+\cdots+S_i}{i+1}$. The payoff of an Asian call option at maturity in the discrete-time model is

$$\max(A(n) - X, 0). \tag{3}$$

Thus the option value under the discrete-time model is

$$e^{-rT}E[\max(A(n) - X, 0)]. \tag{4}$$

Our task is to compute above so that the values converge to the true option values as n increases.

The approximation error for a lattice method can be divided into two types: discretization error and interpolation error. The former refers to the error introduced by discretizing the continuous-time model V in Eq. (2) by the discrete model Eq. (4). The latter indicates the error introduced by additional approximation methods used by pricing algorithms. These approximation methods are needed as it is computationally intractable to evaluate Eq. (4) without them. The discretization error is known to converge to zero at the rate $O(n^{-1})$ [22]. If an efficient approximation method is found to introduce the interpolation error that also converges to zero at rate $O(n^{-1})$, the whole algorithm will converge to the true option value at the same rate.

2.1. Pricing on the CRR lattice

The CRR lattice model is due to Cox et al. [31]. In the CRR lattice model, S_{i+1} equals S_iu with probability p and S_id with probability $1 - p$, where $d < u$. The stock price at time step i that results from j down moves and $i - j$ up moves therefore equals $S_0u^{i-j}d^j$. A 2-time-step CRR lattice is depicted in Fig. 1(a).

We now describe the geometry of the CRR lattice in more detail. Let node $N(i,j)$ stand for the node at time step i reachable from the root with j cumulative down moves. Its associated stock price is $S_0u^{i-j}d^j$. The stock price can move from $N(i,j)$ to $N(i+1,j)$ with probability p and to $N(i+1,j+1)$ with probability $1 - p$. Node $N(i,j)$ can therefore be reached with probability $\binom{i}{j}p^{i-j}(1-p)^j$. See Fig. 1(b) for illustration. For pricing purposes, the probability p for an up move is set to $(e^{r\Delta t} - d)/(u - d)$, where r is the risk-free interest rate, $u = e^{\sigma\sqrt{\Delta t}}$, and $d = 1/u$.

Pricing on the lattice is done by backward induction. The option value at the maturity date can be evaluated directly by Eq. (3). Let (i,j,A) denote the state with an average stock price A (from time step 0 to time step i) at node $N(i,j)$ and $v(i,j,A)$ the corresponding option value. If this stock price moves up to node $N(i+1,j)$ at time step $i+1$, the average stock price becomes $A' \equiv \frac{(i+1)A+Su^{i+1-j}d^j}{i+2}$. If the stock price moves down to node $N(i+1,j+1)$, the average stock price becomes $A'' \equiv \frac{(i+1)A+Su^{i-j}d^{j+1}}{i+2}$. The desired option value $v(i,j,A)$ then equals

$$v(i,j,A) = \mathrm{e}^{-r\Delta t}[p \times v(i+1,j,A') + (1-p) \times v(i+1,j+1,A'')]. \tag{5}$$

The above formula can be applied inductively from time step $n-1$ to time step 0 with $v(0,0,S_0)$ at the root node giving the desired price under the lattice model.

2.2. Interpolation

Although the option value computed by the lattice model converges to the true option value at rate $O(n^{-1})$, how the pricing problem can be solved efficiently poses a challenge. There are $\frac{i!}{(i-j)!j!}$ price paths that reach node $N(i,j)$, and each such path gives rise to a distinct average price (state). The number of states therefore rises dramatically, making the computation via a direct application of Eq. (5) very time consuming. To address this problem, we follow the Hull–White paradigm in lowering the number of states at each node. When state (i,j,A) is missing, its corresponding option value will be estimated by linear interpolation from its two nearest allocated states (i,j,A^-) and (i,j,A^+) via:

$$v(i,j,A) = \frac{A-A^-}{A^+-A^-}v(i,j,A^+) + \frac{A^+-A}{A^+-A^-}v(i,j,A^-), \tag{6}$$

where $A^- < A < A^+$. The term "interpolation error" shall refer to the error arising from estimating $v(i,j,A)$ by linear interpolation.

3. The new $O(n^{2.5})$-time pricing algorithm

Two key techniques are exploited by the algorithm. First, $v(i,j,A)$ can be evaluated directly when A exceeds a certain easily calculated bound. This result helps reduce the state count. Second, a state allocation scheme is developed by applying Lagrange multipliers to minimize the interpolation error. The final state count turns out to be $O(n^{2.5})$, which is also the running time.

3.1. Pruning unnecessary states

Observe that the corresponding price sum for state (ℓ,m,A) is $(\ell+1)A$, as A is the average stock price from time step 0 to time step ℓ. The following theorem states that $v(\ell,m,A)$ can be described by a simple formula when $(\ell+1)A \geqslant (n+1)X$.

Theorem 3.1. *Suppose the price sum associated with state* (ℓ, m, A) *is* $(n+1)X + \epsilon$ *for some* $\epsilon \geqslant 0$. *Then the option value* $v(\ell, m, A)$ *equals*

- $[\epsilon + (n-\ell)S_0u^{\ell-m}d^m]/(n+1)$ when $r = 0$, and
- $\mathrm{e}^{-rT}[\epsilon + S_0u^{\ell-m}d^m\mathrm{e}^{r\Delta t}\frac{1-\mathrm{e}^{(n-\ell)r\Delta t}}{1-\mathrm{e}^{r\Delta t}}]/(n+1)$ when $r > 0$.

Proof. If $r > 0$, the expected value of the future price sum $S_{\ell+1} + S_{\ell+2} + \cdots + S_n$ equals

$$S_0u^{\ell-m}d^m\left[\mathrm{e}^{r\Delta t} + \mathrm{e}^{2r\Delta t} + \cdots + \mathrm{e}^{(n-\ell)r\Delta t}\right] = S_0u^{\ell-m}d^m\mathrm{e}^{r\Delta t}\frac{1-\mathrm{e}^{(n-\ell)r\Delta t}}{1-\mathrm{e}^{r\Delta t}}.$$

The option value $v(\ell, m, A)$ therefore equals

$$\begin{aligned}
&\mathrm{e}^{-rT}E\left[\max\left(A_{\mathrm{avg}}(n) - X, 0\right) | \sum_{i=0}^{\ell} S_i = (n+1)X + \epsilon,\ S_\ell = S_0u^{\ell-m}d^m\right] \\
&= \mathrm{e}^{-rT}E\left[\max\left(\frac{(n+1)X + \epsilon + \sum_{i=\ell+1}^{n} S_i}{n+1} - X, 0\right)\right] \\
&= \mathrm{e}^{-rT}\left[\epsilon + S_0u^{\ell-m}d^m\mathrm{e}^{r\Delta t}\frac{1-\mathrm{e}^{(n-\ell)r\Delta t}}{1-\mathrm{e}^{r\Delta t}}\right]/(n+1).
\end{aligned}$$

The case for $r = 0$ is similar. □

As the value $v(\ell, m, A)$ can be calculated without introducing interpolation errors if $A \geqslant (n+1)X$, a pricing algorithm only needs to evaluate $v(\ell, m, A)$ for $A < (n+1)X$. This improves efficiency, by pruning unnecessary states, and accuracy, for not resorting to interpolation.

3.2. The state allocation scheme

Let $k_{i,j}$ stand for the number of states allocated at node $N(i,j)$. Define k as the average number of states for each node. The total number of states is equal to $\sum_{0\leqslant j\leqslant i\leqslant n} k_{i,j} \approx k(n^2/2)$ as there are approximately $n^2/2$ nodes. The running time is therefore $\mathrm{O}(n^2k)$. Theorem 3.1 says that $v(i, j, A)$ can be easily evaluated if $(i+1)A > (n+1)X$. Thus at node $N(i,j)$, all $k_{i,j}$ states have average stock prices not more than $(n+1)X/(i+1)$. These $k_{i,j}$ states (the average stock prices) shall divide the range $[0, (n+1)X/(i+1)]$ evenly. The difference of the average stock prices of two adjacent states at node $N(i,j)$ is $\frac{(n+1)X}{(i+1)k_{i,j}} \leqslant \frac{nX'}{ik_{i,j}}$, where $X' \equiv 2X$.

The state allocation scheme introduces interpolation error because only $k_{i,j}$ states are allocated for $N(i,j)$ instead of the full $\frac{i!}{j!(i-j)!}$ states and because interpolation formula (6) is employed. The interpolation error can be analyzed as follows. When we calculate $v(i, j, A)$ with formula (5), the option values for the nonexistent states $(i+1, j, A')$ and $(i+1, j+1, A'')$ are estimated by interpolation via Eq. (6). Thus $v(i+1, j, A')$ is interpolated from the values of the two bracketing states $(i+1, j, A'_-)$ and $(i+1, j, A'_+)$, where $A'_- < A' < A'_+$. The interpolation error can be estimated by the Taylor series expansion

$$\frac{(A'_+ - A')(A' - A'_-)}{2}\frac{\partial^2 V(\eta)}{\partial A^2},$$

where $\frac{\partial^2 V(\eta)}{\partial A^2}$ denotes the second partial derivative of the true option value V with respect to the average price A and $\eta \in [A'_-, A'_+]$. We follow [22] in postulating that $\left|\frac{\partial^2 V(\eta)}{\partial A^2}\right|$ is bounded by a constant M. The interpolation error for estimating $v(i+1, j, A')$ is then bounded above by $M(nX'/ik_{i+1,j})^2$. Similarly, the interpolation error for $v(i+1, j+1, A'')$ is at most $M(nX'/ik_{i+1,j+1})^2$. Thus the accumulated interpolation error $\epsilon_a(i, j, A)$ for state (i, j, A) is bounded above by

$$pM(nX'/ik_{i+1,j})^2 + (1-p)M(nX'/ik_{i+1,j+1})^2 + p\epsilon_a(i+1, j, A') \\ + (1-p)\epsilon_a(i+1, j+1, A'').$$

Inductively, the accumulated interpolation error at $(0, 0, S_0)$ is bounded by

$$\epsilon_{\text{int}} \equiv \sum_{1 \leqslant j \leqslant i \leqslant n} \binom{i}{j} p^{i-j}(1-p)^j M(nX'/ik_{i,j})^2 = X'^2 M \sum_{i=1}^{n}\sum_{j=0}^{i}\frac{B(i,j;p)n^2}{i^2k_{ij}^2},$$

where $B(i,j;p) \equiv \binom{i}{j}p^{i-j}(1-p)^j$. To minimize ϵ_{int} subject to the condition $\sum_{1 \leqslant j \leqslant i \leqslant n} k_{i,j} = n^2k/2$, $k_{i,j}$ can be easily shown by the method of Lagrange multipliers to be

$$k_{i,j} = \frac{n^2k}{2} \times \frac{\left[B(i,j;p)/i^2\right]^{1/3}}{\sum_{0 \leqslant m \leqslant l \leqslant n}\left[B(l,m;p)/l^2\right]^{1/3}}.$$

The minimized ϵ_{int} then equals

$$X'^2 M \sum_{1 \leqslant j \leqslant i \leqslant n} \frac{B(i,j;p)n^2}{\frac{i^2n^4k^2}{4} \times \frac{\left[n^2B(i,j;p)/i^2\right]^{2/3}}{\left\{\sum_{0 \leqslant m \leqslant l \leqslant n}\left[n^2B(l,m;p)/l^2\right]^{1/3}\right\}^2}}$$

$$= \frac{4X'^2M}{n^4k^2}\left\{\sum_{1 \leqslant j \leqslant i \leqslant n}\left[n^2B(i,j;p)/i^2\right]^{1/3}\right\}^3$$

$$= \frac{4X'^2M}{n^4k^2}\left\{n^{2/3}\sum_{1 \leqslant i \leqslant n}\left[i^{-2/3}\sum_{1 \leqslant j \leqslant i}B(i,j;p)^{1/3}\right]\right\}^3.$$

A result of Bender [32, p. 489] implies that

$$\sum_{1\leqslant j\leqslant i} B(i,j;p)^{1/3} \leqslant \sum_{1\leqslant j\leqslant i} B(i,j;0.5)^{1/3} \sim (1/2)^{1/3}3^{1/2}\pi^{1/3}i^{1/3} = bi^{1/3},$$

where $b = (1/2)^{1/3}3^{1/2}\pi^{1/3}$. Since

$$n^{2/3}\sum_{1\leqslant i\leqslant n} i^{-2/3} \times bi^{1/3} \sim n^{2/3}\int_0^n bi^{-1/3}\,\mathrm{d}i = (3/2)bn^{4/3}.$$

ϵ_{int} is bounded above by

$$\frac{4X'^2Mb^3(3/2)^3n^4}{n^4k^2} = \frac{4X'^2Mb^3(3/2)^3}{k^2} = \mathrm{O}(k^{-2}).$$

Forsyth et al. argue that the lattice discretization error introduced by discretizing both the time and the stock prices converges at rate $\mathrm{O}(n^{-1})$ [22]. To ensure that the convergence rate of our algorithm is $\mathrm{O}(n^{-1})$, we obtain $\epsilon_{\mathrm{int}} = \mathrm{O}(n^{-1})$ by setting k to be proportional to $n^{0.5}$. As $n^2k = \mathrm{O}(n^{2.5})$, our proposed algorithm runs in time $\mathrm{O}(n^{2.5})$.

4. Numerical Results

When the exercise price of the Asian option is zero, a closed-form solution exists for the Asian option [26]. The simple numerical test in Fig. 2 highlights

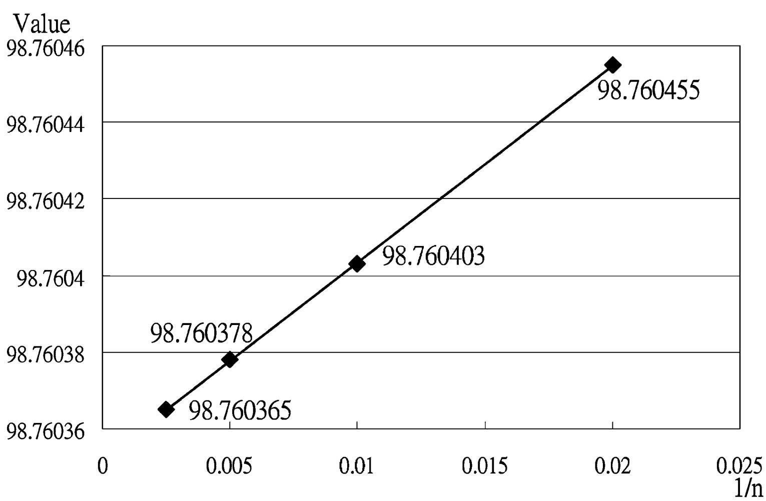

Fig. 2. Estimating the true option value by extrapolation. The stock price is 100, the exercise price is 0, the risk-free rate is 10%, the time to maturity is 0.25 year, and the volatility of the underlying stock is 0.1. The x-axis denotes $1/n$. The y-axis denotes the option value. Four pricing results ($n = 50, 100, 200, 400$) are marked with squares with the calculated option values listed next to the squares. The straight line is computed by linear regression.

that our algorithm converges at rate $O(n^{-1})$. Observe that our prices lie essentially on a straight line. The limiting price when n tends to infinity can be estimated by extrapolation. The extrapolation value of 98.760352 matches the true option value of 98.76035189.

The convergence behaviors of the modified Hull–White method, the PDE method, and our method are compared in Table 1. The time complexities of the first two approaches are $O(n^{3.5})$ and $O(n^3)$, respectively [22]. The number of average states k is set to $250\sqrt{n}$ so that our algorithm uses fewer states in each case than the other two methods. It therefore takes less time as well. All three methods converge to the true option values. But our method achieves the same convergence rate with much less time than the other two methods.

The following numerical tests demonstrate that the extrapolated values obtained by our method are accurate. Zhang provides a very accurate semi-analytical model for pricing the Asian option [14]. Zhang uses this method as the benchmark and compares many different pricing methods [5]. His numerical data are repeated in Table 2 with the extrapolated option values computed by our lattice algorithm added. Both the root-mean-squared errors and the maximum absolute errors of our extrapolated results are much lower than other methods. Interestingly, our method generates much lower pricing errors than other methods when the volatility is large. This is because higher volatility increased the likelihood that the closed-form formula of Theorem 3.1 can be used in pricing.

Table 1
Forsyth's modified Hull–White, PDE, and our methods

	MHW $O(n^{3.5})$		PDE $O(n^3)$		Ours $O(n^{2.5})$	
Time steps	Value	Time (s)	Value	Time (s)	Value	Time (s)
Case 1	$r = 0.1,\ \sigma = 0.1,\ T = 0.25,\ X = 100,\ S = 100$					
50	1.8486	18	1.8478	4.8	1.8487	1
100	1.8501	204	1.8492	55.0	1.8502	7
200	1.8508	2293	1.8503	313.0	1.8509	45
400	1.8512	25918	1.8509	2540.0	1.8512	270
Extrapolation	1.8516		1.8514		1.8516	
	True value: 1.8515 ± 0.0001					
Case 2	$r = 0.1,\ \sigma = 0.5,\ T = 5,\ X = 100,\ S = 100$					
50	28.3899	15	28.3573	6	28.3882	1
100	28.3972	168	28.3842	36	28.3964	7
200	28.4011	1893	28.3952	280	28.4007	45
400	28.4031	21370	28.4003	2278	28.4030	271
Extrapolation	28.4051		28.4054		28.4050	
	True value: 28.40525 ± 0.00015					

MHW is the modified Hull–White method; PDE is the discretized PDE method; Ours is our algorithm; Extrapolation is the extrapolated option values. The results of the modified Hull–White method, the PDE method, and the range of true option values, are based on [22].

Table 2
Comparison with analytical approximations

σ	X	Exact	Extrapolation	AA3	J-TE	PM-J3	PM-J4	CT-GC
0.05	95	8.8088392	8.808871	8.80884	8.80884	8.80884	8.80884	8.80884
	100	4.3082350	4.308312	4.30823	4.30824	4.30822	4.30823	4.30823
	105	0.9583841	0.958609	0.95838	0.95837	0.95841	0.95838	0.95833
0.1	95	8.9118509	8.911908	8.91184	8.91190	8.91175	8.91186	8.91183
	100	4.9151167	4.915249	4.91512	4.91513	4.91514	4.91512	4.91508
	105	2.0700634	2.070162	2.07006	2.06996	2.07025	2.07006	2.06993
0.2	95	9.9956567	9.995679	9.99569	9.99594	9.99550	9.99552	9.99536
	100	6.7773481	6.777354	6.77738	6.77692	6.77819	6.77720	6.77700
	105	4.2964626	4.296472	4.29649	4.29561	4.29791	4.29641	4.29593
0.3	95	11.6558858	11.655841	11.61518	11.65565	11.65663	11.65500	11.65475
	100	8.8287588	8.828706	8.82900	8.82686	8.83183	8.82792	8.82755
	105	6.5177905	6.517738	6.51802	6.51494	6.52237	6.51726	6.51635
0.4	95	13.5107083	13.510619	13.51182	13.50887	13.51308	13.50764	13.50789
	100	10.9237708	10.923669	10.92474	10.91903	10.93043	10.92085	10.92090
	105	8.7299362	8.729839	8.73089	8.72337	8.73968	8.72764	8.72680
0.5	95	15.4427163	15.442573	15.44587	15.43806	15.44623	15.43448	15.43707
	100	13.0281555	13.028020	13.03107	13.01889	13.03880	13.02013	13.02253
	105	10.9296247	10.929477	10.92353	10.91731	10.94583	10.92260	10.92375
RMSE			0.000101	0.00129	0.00434	0.00561	0.00339	0.00268
MAE			0.000225	0.00315	0.01231	0.01621	0.00824	0.00587

Exact is the option value obtained in [14]; Extrapolation is the extrapolated value computed by our method; AA3 is the fourth-order approximation method given in [5]; J-TE is the Taylor expansion method given in [16]; PM-J3 is the shifted lognormal fitting method in [33]; PM-J4 is the shifted arcsinh-normal fitting method in [7]; GT-GC is the continuous limits of the geometric conditioning method given in [34]; RMSE is the root-mean-squared errors. MAE is the maximum absolute error.

5. Conclusion

This paper proposes a new approximation algorithm for pricing Asian options on a lattice. Our algorithm runs in O($n^{2.5}$)-time with the convergence rate O(n^{-1}), which is superior to existing lattice and the related PDE algorithms with the same convergence rate. Our claims are proved rigorously, and numerical results are provided to support the performance and convergence claims.

References

[1] Y.-D. Lyuu, Financial Engineering and Computation: Principles, Mathematics, and Algorithms, Cambridge University Press, Cambridge, UK, 2002.

[2] F. Black, M. Scholes, The pricing of options and corporate liabilities, J. Political Econ. 81 (1973) 637–659.

[3] P. Chalasani, S. Jha, I. Saias, Approximate option pricing, Algorithmica 25 (1999) 2–21.
[4] J.E. Ingersoll, Theory of Financial Decision Making, Rowman & Littlefield, Savage, MD, 1987.
[5] J.E. Zhang, Pricing continuously sampled Asian options with perturbation method, J. Futures Markets 23 (2003) 535–560.
[6] E. Levy, Pricing European average rate currency options, J. Int. Money Finan. 11 (1992) 474–491.
[7] M.A. Milevsky, S.E. Posner, Asian options, the sum of lognormals, and the reciprocal gamma distribution, J. Finan. Quant. Anal. 33 (1998) 409–422.
[8] S.M. Turnbull, L.M. Wakeman, A quick algorithm for pricing European average options, Finan. Quant. Anal. 26 (1991) 377–389.
[9] H. Geman, M. Yor, Bessel processes, Asian options, and perpetuities, Math. Finan. 3 (1993) 349–375.
[10] H. Geman, A. Eydeland, Domino effect, Risk 8 (1995) 65–67.
[11] W.T. Shaw, Modeling Financial Derivatives with Mathematica, Cambridge University Press, Cambridge, U.K., 1998.
[12] J. Abate, W. Whitt, Numerical inversion of Laplace transforms of probability distributions, ORSA J. Comput. 7 (1995) 36–43.
[13] L.C.G. Rogers, Z. Shi, The value of an Asian option, J. Appl. Probab. 32 (1995) 1077–1088.
[14] J.E. Zhang, A semi-analytical method for pricing and hedging continuously sampled arithmetic average rate options, J. Comput. Finan. 5 (2001) 59–79.
[15] M.C. Fu, D.B. Dilip, T. Wang, Pricing continuous Asian options: a comparison of Monte Carlo and Laplace transform inversion methods, J. Comput. Finan. 2 (1998/1999) 49–74.
[16] N. Ju, Pricing Asian and basket options via Taylor expansion, J. Comput. Finan. 5 (2002) 79–103.
[17] P.P. Boyle, M. Broadie, P. Glasserman, Monte Carlo methods for security pricing, J. Econ. Dynam. Control 21 (1997) 1267–1321.
[18] M. Broadie, P. Glasserman, Estimating security price derivatives using simulation, Manage. Sci. 42 (1996) 269–285.
[19] M. Broadie, P. Glasserman, S. Kou, Connecting discrete and continuous path-dependent options, Finan. Stochastics 3 (1999) 55–82.
[20] A.G.Z. Kemna, A.C.F. Vorst, A pricing method based on average asset values, J. Banking Finan. 14 (1990) 113–129.
[21] F.A. Longstaff, E.S. Schwartz, Valuing American options by simulation: a simple least-squares approach, Rev. Finan. Stud. 14 (2001) 113–147.
[22] P.A. Forsyth, K.R. Vetzal, R. Zvan, Convergence of numerical methods for valuing path-dependent options using interpolation, Rev. Deriv. Rech. 5 (2002) 273–314.
[23] T.-S. Dai, Y.-D. Lyuu, Efficient, exact algorithms for Asian options with multiresolution lattices, Rev. Deriv. Rech. 5 (2002) 181–203.
[24] T.-S. Dai, Y.-D. Lyuu, An exact subexponential-time lattice algorithm for Asian options, in: Proc. of the 15th Annual ACM-SIAM Symp. on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, 2004, pp. 703–710.
[25] J. Hull, A. White, Efficient procedures for valuing European and American path-dependent options, J. Deriv. 1 (1993) 21–31.
[26] J. Barraquand, T. Pudet, Pricing of American path-dependent contingent claims, Math. Finan. 6 (1996) 17–51.
[27] T.R. Klassen, Simple, fast and flexible pricing of Asian options, J. Comput. Finan. 4 (2001) 89–124.
[28] P. Ritchken, L. Sankarasubramanian, A.M. Vijh, The valuation of path dependent contracts on the average, Manage. Sci. 39 (1993) 1202–1213.

[29] D. Aingworth, R. Motwani, J.D. Oldham, Accurate approximations for Asian options, in: Proc. of 11th Annual ACM-SIAM Symp. on Discrete Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, 2000, pp. 891–900.

[30] T.-S. Dai, G.-S. Huang, Y.-D. Lyuu, Extremely accurate and efficient algorithms for Asian options with error bounds, in: Proc. of Quantitative Methods in Finance 2002 Conference, 2002.

[31] J. Cox, S. Ross, M. Rubinstein, Option pricing: a simplified approach, J. Finan. Econ. 7 (1979) 229–264.

[32] E.A. Bender, Asymptotic methods in enumeration, SIAM Rev. 16 (1974) 485–515.

[33] S.E. Posner, M.A. Milevsky, Valuing exotic options by approximating the SPD with higher moments, J. Finan. Eng. 7 (1998) 109–125.

[34] G.W.P. Thompson, Fast narrow bounds on the value of Asian options. Working paper, 2000, Center for Financial Research, Judge Institute of Management Science, University of Cambridge.